Javascript

Session 2: les blocs

Thibault Clérice
@ponteineptique (Twitter et Github)
contact@algorythme.net

Les erreurs : Try.. catch...

- Système de bloc orienté pour empêcher des erreurs de faire crasher un script

```
// Situation avec Erreur
try {
    jenexistepas("ici");
} catch (error) {
    console.log(error);
} finally {
    console.log("still running !");
}
```

```
// Situation sans erreur
try {
    console.log("Hey");
} catch (error) {
    console.log(error);
} finally {
    console.log("still running !");
}
```

If, else, else if

- Un bloc suivant un if est exécuté si les conditions sont “true”
  - If (x < 5 && x > 2) { ? }
    - 3
    - 2
    - 4
    - 5
- Else if exécuté si aucun des blocs qui le précèdent n’est lancé et que ses conditions sont “true”
- Else est exécuté si aucun des blocs qui le précèdent n’est lancé

```
var heure = 10;

if (heure < 10) {
    greeting = "Café ?";
} else if (time < 20) {
    greeting = "Bonjour";
} else {
    greeting = "Bonne nuit";
}
```

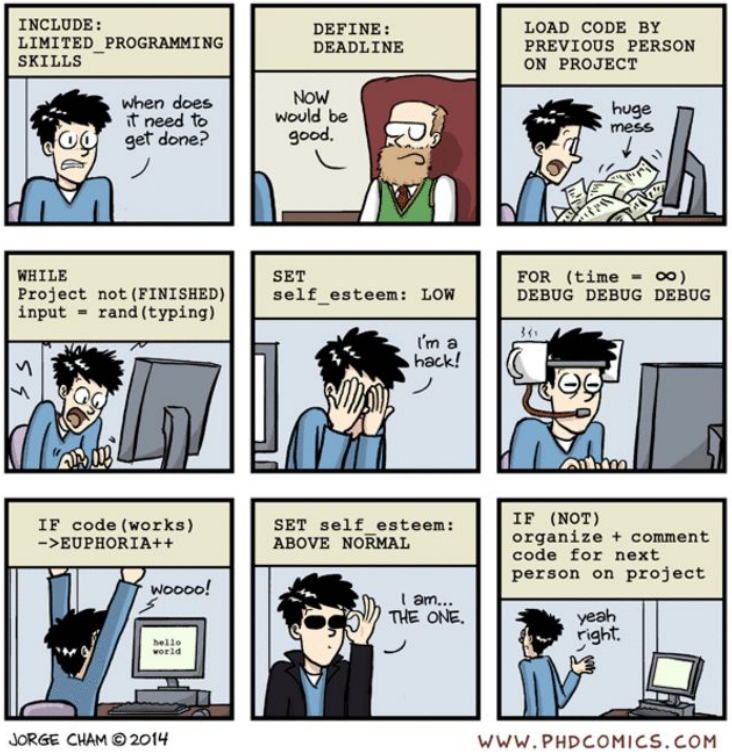
Switch

- Switch ne teste qu'une valeur
- Possède une action par défaut

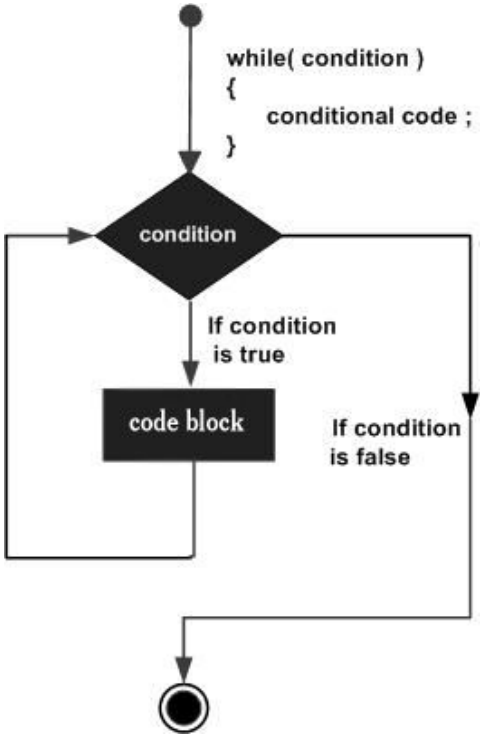
```
var page = "something";

switch (page) {
    case "Accueil":
        console.log("Vous avez selectionne Accueil");
        break;
    case "A propos de":
        console.log("Vous avez selectionne A propos de");
        break;
    case "Bulletin":
        console.log("Vous avez selectionne Bulletin");
        break;
    case "Identification":
        console.log("Vous avez selectionne Identification");
        break;
    case "Liens":
        console.log("Vous avez selectionne Liens");
        break;
    default:
        console.log("Cherchez mieux");
}
```

Les boucles

- Deux grands type de boucles
  - For
    - A sous-divisé en trois
  - While
- Moins commun
  - “Do”



while



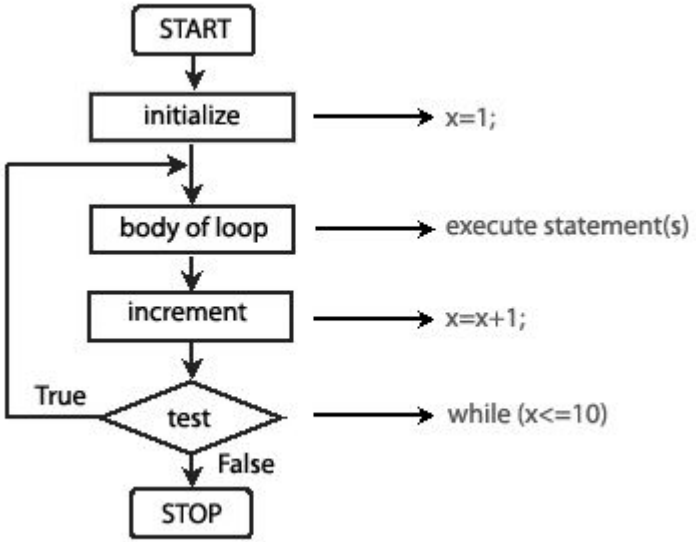

```
var n = 0;
var x = 0;
while (n < 3) {
  n++;
  x += n;
  console.log(x);
}
```

For (basique)

for ([expressionInitiale]; [condition]; [expressionIncrément]) {

  instruction

}




```
ecole = [
    "EPHE",
    "ENC",
    "P4",
    "EHESS"
];
for (var i = 0; i < ecole.length; i++) {
    if (ecole[i] == "ENC") {
       console.log("Index de l'Ecole : " + i);
    }
    console.log("Regarder le " + i);
}
```

For … in ...

Itérer sur l'ensemble des propriétés énumérables d'un objet

- Utile pour les tableaux
- Utile pour les objets

```
voiture = {};
voiture["fabricant"] = "Ford";
voiture["modèle"] = "Mustang";

for (key in voiture) {
  console.log(key);
}
```

For … In ...

```
Texte = {
    "meta":"line",
    "author":"aeschylus",
    "credit":"OpenGreekAndLatin",
    "urn":"urn:cts:greekLit:tlg0085.tlg007.opp-grc3-simple",
    "text":{
        "0":"Πρῶτον μὲν εὐχῇ τῇδε πρεσβεύω θεῶν",
        "1":"τὴν πρωτόμαντιν Γαῖαν: ἐκ δὲ τῆς Θέμιν,",
        "2":"ἣ δὴ τὸ μητρὸς δευτέρα τόδ᾽ ἔζετο",
        "3":"μαντεῖον, ὡς λόγος τις: ἐν δὲ τῷ τρίτῳ",
    }
};
```

A l’aide de deux boucles,
afficher chacune des métadonnées
puis chacun des passages

For … of ...

Itérer sur l'ensemble des propriétés énumérables d'un objet

- Utile pour les tableaux
- Utile pour les objets

```
// Initialisé comm un Array
arr = [3, 5, 7];
// Ajouter une propriété Object
arr.toto = "coucou";

for (i in arr) {
  console.log(i); // affiche 0, 1, 2, "toto" dans la console
}

// Toto ne fait pas parti
for (i of arr) {
  console.log(i);
  // affiche 3, 5, 7 dans la console
  // puisque toto ne fait pas  partie des propriétés array
}
```

Break

Break permet d'arrêter une boucle

Souvent utilisé avec des conditions dans des boucles longues ou avec des données inconnues

```
var a = ["ENC", "EPHE", "Paris IV"];
var valeurTest = "EPHE";

for (i = 0; i < a.length; i++) {
  if (a[i] == valeurTest) {
    break;
  }
  console.log(a[i]);
}
```

Ecrire des fonctions

```
var nom_de_function = function(argument1, argument2) {

    // faire ce que l'on veut ici
    Variable = 1;
    //potentiellement renvoyer une valeur :
    return variable;
}
```

Bonus : Changer des données de type

Conversion en type	Fonction à utiliser
`Int (entier)`	parseInt()
`Boolean (booléen)`	Boolean()
`Float (décimaux)`	parseFloat()
`String`	String()
Similaire	
`String -> Array`	.split()
`Array -> String`	`.join();`

```
var virgile = "Arma virumque cano,
Troiae qui primus ab oris";

console.log(virgile.split(" "));
console.log(
  virgile.split(",").join(" : ")
);
```

Ecrire des fonctions : exercice

Ecrire une fonction dans fonction js se nommant “trouverlemot” prenant un paramètre mot et un dictionnaire de textes (cf. data.js).

Utiliser console.log pour afficher l’identifiant que la phrase possède.

Retourner une liste des valeurs

Vous aurez besoin :

- De boucles
- De trouver sur internet la documentation pour trouver une chaine dans une autre chaîne
- Trouver comment ajouter un élément a une liste

Essayer avec les mots :

Catiline, judge, Sulla